# A METHOD AND APPARATUS FOR PROVIDING MULTI-PROTOCOL, MULTI-STAGE, REAL-TIME FRAME CLASSIFICATION

## CROSS-REFERENCE TO OTHER PATENT APPLICATIONS

The following co-pending patent applications of common assignee contains some common disclosure:

"System And Method For Providing Transformation Of Multi-Protocol Packets In A Data Stream," Attorney Docket No. 1305.1-US-01, filed concurrently herewith, which is incorporated herein by reference in its entirety;

"System And Method For Policing Multiple Data Flows And Multi-Protocol Data Flows," Attorney Docket No. 1305.6-US-01, filed concurrently herewith, which is incorporated herein by reference in its entirety;

"System And Method For Hierarchical Policing Of Flows And Subflows Of A Data Stream," Attorney Docket No. 1305.10-US-01, filed concurrently herewith, which is incorporated herein by reference in its entirety.

## BACKGROUND OF THE INVENTION

### 1. Field of the Invention.

This invention relates in general to a method and apparatus for handling packets, and more particularly to a method and apparatus for providing multi-protocol, multi-stage, real-time frame classification.

### 2. Description of Related Art.

Standards-based LAN systems work reasonably well at transfer rates up to about 100 Mbps. At transfer rates above 100 Mbps, providing the processing power required by a packet switch interconnecting a group of networks becomes economically challenging for the performance levels desired. This difficulty in economically "scaling up" performance is beginning to cause restrictions in some user's planned network expansions. Also, today's data networks do not provide network managers with enough control over bandwidth allocation and user access.

Next generation networks are expected to support "multimedia" applications with their much greater bandwidth and real-time delivery requirements. The next generation networks should also have the ability to dynamically adjust the network so that it can guarantee a predetermined amount of bandwidth for the requested service level agreement. Additionally, it is desirable to provide access, performance, fault tolerance and security between any specified set of end systems as directed by the network's manager.

One of the biggest opportunities for service providers today is to provide IP–based internetworking services to meet the exponential growth in demand from both business and residential customers. For example, voice and video based multimedia applications are expected to become a significant portion of the Internet. However, support for multimedia applications in the current Internet is at its initial stages. To fulfill the promise of remote work styles and B2C (Business-to-Consumer) e-commerce, broadband access for small offices, home offices and residences is critical. In today's information-based society, many individuals desire remote data connectivity to an office or remote data site. Remote individuals desire remote and transparent connectivity to the corporate office or a remote data site, including connectivity to the corporate office local area network (LAN).

Broadband systems are being developed and implemented to provide higher capacities, more efficient use of bandwidth, and the ability to integrate voice, data, and video communications. The number and type of communication services has also been rapidly expanding, including the above-mentioned "multimedia" services such as video teleconferencing, video/movies on demand and the like.

While broadband access is becoming more common, there are many competing technologies for delivering broadband access. For example, there are a number of standards used in digital telecommunications, including TCP/IP, Ethernet, HDLC, ISDN, ATM, X.25, Frame Relay, Digital Data Service, FDDI (Fiber Distributed Data Interface), T1, xDSL, Wireless, Cable Modems, and Satellite among others. Many of these standards employ different packet and/or frame formats. The term "frame" generally refers to encapsulated data at OSI layer 2, including a destination address, control bits for flow control, the data or payload, and CRC (cyclic redundancy check) data for error checking. The term "packet" generally refers to encapsulated data at OSI layer 3. However, in the present application, the term packet and frame and cell will be used interchangeably.

In general, a packet format or frame format refers to how data is encapsulated with various fields and headers for transmission across a network. For example, a data packet typically includes an address destination field, a length field, an error correcting code (ECC) field or cyclic redundancy check (CRC) field, as well as headers and trailers to identify the beginning and end of the packet. The terms "packet format" and "frame format", also referred to as "cell format", are generally synonymous.

In order for a router or gateway to be able to interface between communication systems employing different packet or frame formats, the node element, such as the router or gateway, is required to perform a packet conversion to convert the data from a first packet format used by the first communication system to a second packet format used by the second communication system. As the number of possible packet formats or types increases, the amount of logic required to convert between these different packet formats also increases.

Furthermore, the emergence of high speed networking technologies, e.g., ATM cell-based technology, xDSL, cable modem technology and Gigabit Ethernet, makes possible the integration of multiple types of traffic like speech, video and data over the same communication network. The communication circuits which may be shared in such network include transmission lines, program controlled processors, nodes or links, and data or packet buffers. An access node in such network should thus be designed for supporting the access of the user existing communication equipment with their corresponding protocols as well as for accommodating new equipment. Accordingly, it is essential to know the different requirements of each traffic in order to optimize the different processes.

Real-time traffic has more constraining requirements than non-real-time ones, i.e., end-to-end delay and jitter. It is necessary to be able to give priority to the real-time packets in order to minimize these delays. Meanwhile, the packet loss must be guaranteed both for real-time and non-real-time applications that have reserved bandwidth in the network while it is not mandatory for non-reserved type of traffic.

Therefore, it is important to provide the network components (i.e. nodes or links) with mechanisms that control the priority of the packets and process them so that the desired quality of service (QOS) to their corresponding connections is guaranteed. It is also important to offer the same service to connections having the same QOS requirements by providing them with a fair share of the network transmission capacities.

As can be seen, the technological convergence of computer and communication networks has led to more complex transmission of data, voice, images etc. Depending on the network, various protocols are hierarchically ordered, resulting in a vertical stack of protocols. Each of these protocols interact with the adjacent ones to organize the information exchange and transmission between remote systems, such as host computers. If an application program, for example, which runs on a first system requires the use of data of a second system, an exchange of information takes place. When the second system receives a request to send specific information, this information has to be transmitted from the highest protocol level, e.g., the application layer, down through all lower protocol levels prior to being sent along the physical link. Each protocol layer adds its own layer-specific connection information to data packets containing the request information that are received from the higher layer.

Thus, a communication connection between two systems is defined in a packet header, hereinafter referred to as protocol header, by the aggregate of fields carrying connection information of the vertical protocol stack. Nevertheless, when receiving a data stream made up of data packets at a receiver site, prior to forwarding, routing, multiplexing or compressing the data packets, the protocol header has to be scanned to extract information to at least identify the connection information.

A fundamental function in processing packets in networking communications is filtering. Filtering is the process of applying a set of rules to an incoming packet in order to determine its forwarding characteristics. Advanced frame identification and/or marking may be used to identify the entire frame composition layer by layer. The rules that are applied to perform the filtering can vary. For example, plural criteria may be used in a given table lookup. Another example uses the results of a one table lookup with certain packet protocol criteria to generate subsequent table lookups. Current methods used to obtain this type of filtering are implemented in software and thereby do not scale with the bandwidth in today's networks. Furthermore, these operations consume a considerable amount of time in the protocol processing, in particular when dealing with many connections, e.g., in a server, or when processing multimedia data streams. The result of this is cumbersome and conventional filtering applied to the high-speed networks leads to network degradation.

Hardware implementation of a routing table for the translation of packet identifiers into an appropriate physical output link has been described in "Putting Routing Tables in Silicon", T.-B. Pei and C. Zukowski, IEEE Network Magazine, January 1992, pp. 42-50. This approach is mainly characterized in that a Content Addressable Memory (CAM) is employed to match connection information in the header of a single protocol. In addition, the advantages and disadvantages of CAMs versus conventional Random Access Memories (RAM), used to store routing information, have been evaluated by Pei and Zukowski.

In addition to the above-mentioned problenıs, another problem associated with using a CAM to match connection information in the header. To make filtering decisions, a CAM table is built. The CAM table contains search words. The table may contain fields for the IP source, the type of service, the TCP source port, etc. However, as search words are built, the amount of memory required explodes exponentially.

Neither of the two systems above, both of them relating to the solution of sub-problems, nor the known software approaches allow fast processing of multiple protocols. A wide variety of communication protocols exist, but all tend to fall into one of the following groups: LAN protocols, WAN protocols, network protocols, and routing protocols. LAN protocols operate at the network and data link layers of the OSI model and define communication over the various LAN media. WAN protocols operate at the lowest three layers of the OSI model and define communication over the various wide-area media. Routing protocols are network-layer protocols that are responsible for path determination and traffic switching. Finally, network protocols are the various upper-layer protocols that exist in a given protocol suite. The processing of protocol headers and the recognition of different protocol types in real time is a very complicated and difficult undertaking. In almost all network systems, header processing is still a major CPU-cycle (Central Processor Unit) consuming activity.

It can be seen that there is a need for a method and apparatus for providing multi-protocol, multi-stage, real-time frame classification.

## SUMMARY OF THE INVENTION

To overcome the limitations in the prior art described above, and to overcome other limitations that will become apparent upon reading and understanding the present specification, the present invention discloses a method and apparatus for providing multi-protocol, multi-stage, real-time frame classification.

The present invention solves the above-described problems by providing an advanced level of data extraction across various frame protocols. Longest prefix match searches and/or direct lookup searches are supported. Moreover, conditional extractions, instruction branching, and multi-stage processing are all performed in real time.

A method in accordance with the principles of the present invention includes performing preliminary multi-protocol frame classification for incoming frames, processing at least the incoming frame and the preliminary multi-protocol frame classification to provide parsing instructions, and providing multi-stage parsing of the incoming frame according to the parsing instructions to generate search results presenting information about the incoming frame.

Other embodiments of a method in accordance with the principles of the invention may include alternative or optional additional aspects. One such aspect of the present invention is that the performing preliminary multi-protocol frame classification further includes resolving a composition for the incoming frame.

Another aspect of the present invention is that the processing further includes storing an incoming frame in a frame buffer, generating the parsing instructions for the incoming frame based upon the multi-protocol frame classification and aligning the generated parsing instructions with a frame selected from the incoming frames stored in the frame buffer.

Another aspect of the present invention is that the providing multi-stage parsing of the incoming frame further includes reading at least a portion of the incoming frame from a frame buffer, interpreting the parsing instructions and executing macros for performing parsing functions on at least the portion of the incoming frame using the interpreted parsing instructions.

Another aspect of the present invention is that the providing multi-stage parsing of the incoming frame further includes generating extraction instructions in response to the at least incoming frame and the preliminary multi-protocol frame classification, using the extraction instructions to produce search queries, and applying the search queries against a memory to produce the search results.

Another aspect of the present invention is that the using the extraction instructions further comprises conditional extraction.

Another aspect of the present invention is that conditional extraction comprises determining whether extracted bits of data from the packet meet a predetermined criteria.

Another aspect of the present invention is that the providing multi-stage parsing of the incoming frame further includes reprocessing the search results to generate refined parsing instructions and parsing of the incoming frame according to the refined parsing instructions to generate refined search results, the refined search results providing greater information about the incoming frame.

Another aspect of the present invention is that the processing at least the incoming frame and the preliminary multi-protocol frame classification further includes processing the search results, the incoming frame and the preliminary multi-protocol frame classification to provide refined parsing instructions.

Another aspect of the present invention is that the parsing instructions includes a location identifier and an associated instruction to be applied to the frame at the position identified by the location identifier.

Another aspect of the present invention is that the frame includes one of a plurality of frame protocols, the performing preliminary multi-protocol frame classification further including classifying a protocol identity.

Another aspect of the present invention is that the performing preliminary multi-protocol frame classification further includes identifying each layer protocol and writing the frame into memory along with a protocol layer code.

Another aspect of the present invention is that providing multi-stage parsing of the incoming frame further includes using the protocol layer code as a start vector into an instruction memory, the instruction memory containing instructions for parsing the frame and pointers to access selected words of the frame in a frame buffer.

Another aspect of the present invention is that the performing preliminary multi-protocol frame classification further includes identifying each layer protocol and writing the frame into memory along with a protocol layer code, the method further including processing the instructions from the instruction memory, resolving the pointers according to the protocol layer table and aligning the frame data with the instruction words.

Another aspect of the present invention is that the search results are fed back to an instruction controller to be used as data that a parsing engine can parse through a data pipe.

Another aspect of the present invention is that the memory includes a content addressable memory and the search queries are used to search the content addressable memory.

Another aspect of the present invention is that the memory includes a static random access memory and the search queries are indexed directly to a static random access memory.

Another aspect of the present invention is that the method further includes providing a parser instruction set in the instruction memory to define microcode used for generating the parser instructions.

Another aspect of the present invention is that the method further includes forwarding the frame and the search results.

Another aspect of the present invention is that the search results include the frame classification.

Another aspect of the present invention is that the search results are fed back to an instruction controller to be used as a start vector for subsequent processing stages.

Another aspect of the present invention is that providing multi-stage parsing of the incoming frame further comprises performing conditional branching, the conditional branching allowing specific instructions to be performed based upon satisfaction of a predetermined criteria.

In another embodiment of the present invention, a multi-protocol, multi-stage, real-time frame classifier is provided. The multi-protocol, multi-stage, real-time frame classifier includes a preliminary multi-protocol frame composition analyzer for performing preliminary multi-protocol frame classification for incoming frames, a parsing instruction generator for processing at least the incoming frame and the preliminary multi-protocol frame classification to provide parsing instructions and a multi-stage parsing engine for providing multi-stage parsing of the incoming frame according to the parsing instructions to generate search results presenting information about the incoming frame.

Another aspect of the present invention is that the preliminary multi-protocol frame classifier resolves a composition for the incoming frame.

Another aspect of the present invention is that the parsing instructions generator processing further includes a frame buffer for storing an incoming frame, an instruction generator for creating the parsing instructions for the incoming frame based upon the multi-protocol frame classification and a data pipe for aligning the generated parsing instructions with a frame selected from the incoming frames stored in the frame buffer.

Another aspect of the present invention is that the multi-stage parsing engine reads at least a portion of the incoming frame from a frame buffer, interprets the parsing instructions and executes macros for performing parsing functions on at least the portion of the incoming frame using the interpreted parsing instructions.

Another aspect of the present invention is that the multi-stage parsing engine generating extractions instructions in response to the at least incoming frame and the preliminary multi-protocol frame classification.

Another aspect of the present invention is that the multi-stage parsing engine performs conditional extractions using the extraction instructions.

Another aspect of the present invention is that conditional extraction comprises determining whether extracted bits of data from the packet meet a predetermined criteria.

Another aspect of the present invention is that the multi-stage parsing engine further includes an extractor for using the extraction instructions to produce search queries and a memory for applying the search queries against to produce the search results.

Another aspect of the present invention is that the multi-protocol, multi-stage, real-time frame classifier further includes a feedback path for allowing the search results to be reprocessed to generate refined parsing instructions that are used by the parsing engine to generate refined search results, the refined search results providing greater information about the incoming frame.

Another aspect of the present invention is that the  parsing instructions generator processes the search results, the incoming frame and the preliminary multi-protocol frame classification to provide refined parsing instructions.

Another aspect of the present invention is that the parsing instructions include a location identifier and an associated instruction to be applied to the frame at the position identified by the location identifier.

Another aspect of the present invention is that the frame includes one of a plurality of frame protocols, the preliminary multi-protocol frame classifier classifying the incoming frame to identify a protocol identity.

Another aspect of the present invention is that the preliminary multi-protocol frame classifier identifies each layer protocol and writes the frame into memory along with a protocol layer code.

Another aspect of the present invention is that the multi-stage parsing engine uses the protocol layer code as a start vector for an instruction memory, the instruction memory containing instructions for parsing the frame and pointers to access selected words of the frame in a frame buffer.

Another aspect of the present invention is that the parsing instructions generator identifies each layer protocol and writes the frame into memory along with a protocol layer code and wherein the multi-stage parsing engine processes the instructions from the instruction memory, resolves the pointers according to the protocol layer table and aligns the frame data with the instruction words.

Another aspect of the present invention is that the search results are fed back to the parsing instructions generator to be used as data that a parsing engine can parse through a data pipe.

Another aspect of the present invention is that the memory includes a content addressable memory and the search queries are used to search the content addressable memory.

Another aspect of the present invention is that the memory includes a static random access memory and the search queries are indexed directly to a static random access memory.

Another aspect of the present invention is that the parsing instructions generator provides a parser instruction set in an instruction memory to define microcode used for generating the parser instructions.

Another aspect of the present invention is that the multi-stage parsing engine forwards the frame and the search results.

Another aspect of the present invention is that the search results include the frame classification.

Another aspect of the present invention is that the parsing instructions generator further includes an instruction controller and wherein the search results are fed back to the instruction controller to be used as a start vector for subsequent processing stages.

Another aspect of the present invention is that the multi-stage parsing engine performs conditional branching, the conditional branching allowing specific instructions to be performed based upon satisfaction of a predetermined criteria.

In another embodiment of the present invention, an article of manufacture comprising a program storage medium readable by a computer is provided. The medium tangibly embodies one or more programs of instructions executable by the computer to perform a method for parsing frames in a network flow, wherein the method includes performing preliminary multi-protocol frame classification for incoming frames, processing at least the incoming frame and the preliminary multi-protocol frame classification to provide parsing instructions and providing multi-stage parsing of the incoming frame according to the parsing instructions to generate search results presenting information about the incoming frame.

In another embodiment of the present invention, another method for handling packets is provided. The method including performing preliminary multi-protocol frame classification for incoming frames, processing at least the incoming frame and the preliminary multi-protocol frame classification to provide parsing instructions and providing parsing of the incoming frame according to the parsing instructions to generate search results presenting information about the incoming frame.

In another embodiment of the present invention, another method for handling packets is provided. The method includes processing at least the incoming frame to provide parsing instructions and providing multi-stage parsing of the incoming frame according to the parsing instructions to generate search results presenting information about the incoming frame.

In another embodiment of the present invention, another article of manufacture comprising a program storage medium readable by a computer is provided. The medium tangibly embodies one or more programs of instructions executable by the computer to perform a method for parsing frames in a network flow, wherein the method includes performing preliminary multi-protocol frame classification for incoming frames, processing at least the incoming frame and the preliminary multi-protocol frame classification to provide parsing instructions and providing parsing of the incoming frame according to the parsing instructions to generate search results presenting information about the incoming frame.

In another embodiment of the present invention, another article of manufacture comprising a program storage medium readable by a computer is provided. The medium tangibly embodies one or more programs of instructions executable by the computer to perform a method for parsing frames in a network flow, wherein the method includes processing at least the incoming frame to provide parsing instructions and providing multi-stage parsing of the incoming frame according to the parsing instructions to generate search results presenting information about the incoming frame.

In another embodiment of the present invention, another multi-protocol, multi-stage, real-time frame classifier is provided. The multi-protocol, multi-stage, real-time frame classifier including a preliminary multi-protocol frame composition analyzer for performing preliminary multi-protocol frame classification for incoming frames, a parsing instruction generator for processing at least the incoming frame and the preliminary multi-protocol frame classification to provide parsing instructions and a parsing engine for providing multi-stage parsing of the incoming frame according to the parsing instructions to generate search results presenting information about the incoming frame.

In another embodiment of the present invention, another multi-protocol, multi-stage, real-time frame classifier is provided. The multi-protocol, multi-stage, real-time frame classifier includes a parsing instruction generator for processing an incoming frame to provide parsing instructions and a multi-stage parsing engine for providing multi-stage parsing of the incoming frame according to the parsing instructions to generate search results presenting information about the incoming frame.

These and various other advantages and features of novelty which characterize the invention are pointed out with particularity in the claims annexed hereto and form a part hereof. However, for a better understanding of the invention, its advantages, and the objects obtained by its use, reference should be made to the drawings which form a further part hereof, and to accompanying descriptive matter, in which there are illustrated and described specific examples of an apparatus in accordance with the invention.

## BRIEF DESCRIPTION OF THE DRAWINGS

Referring now to the drawings in which like reference numbers represent corresponding parts throughout:

Fig. 1 illustrates a block diagram for a network device that includes a multi-protocol route/flow classifier with sophisticated policing engines and packet transformation capability;

Fig. 2 is a block diagram of an input frame device according to the present invention;

Fig. 3 illustrates a block diagram of an apparatus for providing multi-protocol, multi-stage, real-time frame classification according to the present invention;

Fig. 4 illustrates a detailed block diagram of an apparatus for providing multi-protocol, multi-stage, real-time frame classification according to the present invention;

Fig. 5 illustrates a flow chart representing the processes performed by the apparatus for providing multi-protocol, multi-stage, real-time frame classification;

Fig. 6 illustrates the arrangement of the dual port frame buffer; and

Fig. 7 illustrates a flow chart showing the processes of the detailed block diagram of the apparatus for providing multi-protocol, multi-stage, real-time frame classification with reference to Fig. 4 according to the present invention.

## DETAILED DESCRIPTION OF THE INVENTION

In the following description of the exemplary embodiment, reference is made to the accompanying drawings which form a part hereof, and in which is shown by way of illustration the specific embodiment in which the invention may be practiced. It is to be understood that other embodiments may be utilized as changes may be made without departing from the scope of the present invention.

The present invention provides a method and apparatus for providing multi-protocol, multi-stage, real-time frame classification. Advanced level of data extraction is provided across various frame protocols without imposing a performance penalty. Longest prefix match searches and/or direct lookup searches are supported. Moreover, conditional extractions, instruction branching, and multi-stage processing are all performed in real time.

Fig. 1 illustrates a block diagram for a network device 100 that includes a multi-protocol route/flow classifier with sophisticated policing engines and packet editing capability. The network device 100 provides advanced services at 10 Gbps, at least, by tightly coupling parsing, policing, and packet editing. This coupling allows the device to perform dynamic packet editing for QoS based on the current flow state and also effectively handles dynamic header processing such as required by MPLS routers. A physical layer interface 110 provides access to a framer 112 on the ingress. A input frame device 120 includes a parsing engine for handling differentiated services and providing packet analysis, a rate policing engine, and editor functionality. Tightly coupled parsing, policing, and packet editing allows the collective device to perform dynamic packet editing for quality of service based on the current flow state and also effectively handles dynamic header processing such as required by MPLS routers. A session or flow is defined as a series of IP datagrams that share a common destination IP address and traffic-engineering characteristics.

The input frame device 120 provides differentiated services, routing, and policing and provides them to a switch fabric and interface 130. On the egress side, the egress frame device 140 provides differentiated services traffic management. The egress frames are then provided to the framer 112 and then onto the physical layer interface 110. A processor 150 controls at least the input 120 and egress 140 frame devices.

Fig. 2 is a block diagram of an input frame device 200 according to the present invention. In Fig. 2, the input frame device includes a classifier 210 that receives incoming frames 212. The classifier builds queries (search words) to be search against either a CAM 214 or indexed directly to a SRAM 216. A policer 220 is provided to ensure flow conformance to a maximum allowed peak rate and a contractually obliged committed rate for flows, e.g., DiffServ IP and MPLS. The policer 220 includes an SRAM 222 for storing a drop policy for each connection. An editor 230 performs the last operations to the packet before being forwarded as outgoing frames 232. The editor 230 may fetch editor instructions and support policing results. An external SRAM 234 is provided for storing editor instructions. A processor interface allows for control of the classifier 210, policer 220 and editor 230 via the processor 242. In addition, a memory 244 may be provided to the processor interface 240 also.

Fig. 3 illustrates a block diagram 300 of an apparatus for providing multi-protocol, multi-stage, real-time frame classification according to the present invention. In Fig. 3, a preliminary frame composition analyzer 310 is provided to produce the preliminary multi-protocol frame classification 312 for incoming frames 314. A parsing instructions generator 320 is provided for processing at least the incoming frame 314 and the preliminary multi-protocol frame classification 312 to provide parsing instructions 322. A multistage parsing engine 330 receives at least a portion of the incoming frame 314 and the parsing instructions 322 to produce the search results 332.

The task of packet parsing involves identification of the fields in the packet. Field identification does not present a problem in the case of Layer 2/Layer 3/Layer 4 applications where the field offsets are known either with respect to the start of the packet or are dependent on the contents of another field. However, content-aware classification rules often require fields located within the data packet payload. In this case, the off-sets of fields may not be known apriori and the rule must encode the parsing and classification information.

The multistage parsing engine 330 provides multi-stage parsing of the incoming frame 314 according to the parsing instructions 322 to generate search results that presents information about the incoming frame 314. The search results 332 may be fedback 350 to the parsing instructions generator 320 for reprocessing of the search results 332 to generate parsing instructions 322 that are refined. Then, the multistage parsing engine 330 parses at least a portion of the incoming frame according to the refined parsing instructions 322 to generate search results

330 that are refined. The refined search results 330 provided greater information about the incoming frame 314.

Accordingly, the parsing engine 330 performs layer classification and tagging via a search/lookup engine. The parser 330 includes bitwise extraction from layer and offset and limited operators and masks. The entire frame is available for parsing and extraction. The parser generates, for example, up to four 72-bit search keys that can be concatenated for up to 288 bit searches and is capable, for example, of 2 stage lookup. For example, the parsing engine 330 may perform 512 programmed extractions on the first search and 512 for second stage extractions.

A microcode controlled programmable sequencer implementation drives the parsing engine. The parsing engine 330 is programmable to build search words. The incoming frame 314 is parsed to flexibly generate the search words from data anywhere within the frame or cell. The microcode instruction set contains specialized instructions defined to perform powerful and flexible operations such as EXTRACT, MASK, NOT, AND, OR,XOR, and Range. The instructions and associated data are executed from a dynamically loadable code space.

To prevent table explosion, a bit may be set as a key to indicate whether a field in a packet is within a predetermined range. For example, when a packet is analyzed, if the TCP source port is within a predetermined range, a bit is set to represent the TCP source port rather than building a search word having the typical 16 bits for the TCP source port. Accordingly, the CAM is used more effectively because less memory is needed to build the search words. The results of the CAM search are applied to the SRAM and returned to the result buffer. The parser 330 builds a search word by extracting bits of data from the packet and prepending bits representing the search type.

The parsing engine 330 parses the incoming frame 314 by extracting data into search words having various combinations of bits, e.g., 32, 64, 128, or 256 bits. Multiple searches may be performed, e.g., up to four searches may be issued to the CAM for Longest Prefix Matching or Direct SRAM Lookup (SRAM). For the Direct SRAM Lookup searches, the upper bits are truncated to match the width of the SRAM address space. The two words within the search word also have a corresponding eight bit search type. Given a specified extraction criteria, the appropriate microcode for the parser is provided. The protocol layer, extract data start offset, bits to be extracted, search word and location for extracted bits, N Bits of search type and SRAM or CAM Lookup parameters must be specified before the appropriate microcode can be generated. Optional parameters include data operations, e.g., greater than, less than, equal, NOT, AND, OR, XOR, and Range, that returns a single true/false bit that can be inserted into the search word. The true/false bit can also be used as a conditional branch, as will be explained with reference to Fig. 4. Optional parameters also includes a bit mask and operand(s) and branch instructions. Multiple microcode instructions can be concatenated into the same search words. The last instruction of a multiple instruction operation must indicate the extraction is complete.

The parsing engine 330 supports at least Longest Prefix Matching (LPM) lookups for forwarding table entries, MPLS label-based routing using direct lookup SRAM tables, ATM cell Virtual Path Identifier(VPI)/Vertical Connector Identifier (VCI) mapping using direct lookup SRAM tables and Layer 3 lookups of source address (SA), destination address (DA), and type-of-service (TOS) fields. The parsing engine 330 may perform accesses to CAM or direct lookup SRAM. The parsing engine 330 supports a multi-stage memory lookup where the Connection ID resulting from one lookup is utilized to create the query word for a subsequent lookup.

Fig. 4 illustrates a detailed block diagram 400 of an apparatus for providing multi-protocol, multi-stage, real-time frame classification according to the present invention. A preprocessor 410 classifies the type of frame 412 received. The preprocessor 410 may be programmed to identify any protocol in any OSI layer. The preprocessor 410 identifies each layer protocol using a multistage algorithm coupled with a CAM/SRAM 414 for resolving protocols, etc.

Fig. 6a illustrates the arrangement of the dual port memory buffer 600. In Fig. 6a, the memory buffer contains a first frame 610 and 2-N frames 620. The first frame 610 is shown having several segments. The first segment is the protocol header 632 followed by the Layer 2 header 634, the Layer 2.5 header 636, the layer 3 header 638 and the layer 4 header 640. The layer address table 650 in Fig. 6b provides the layer base addresses 653 which provide an indication of the address for each layer along with the search results 656, the end-of-frame 654, and other programming 658.

Referring again to Fig. 4, the frame is then written into the dual port memory buffer 416 along with the result of the preprocessor, the protocol layer code. The protocol layer code and layer 653 are stored in the protocol header 632. The dual port memory buffer 416 includes port A 418 and port B 420. Each layer boundary is marked as to where it resides, e.g., an N-bit code is stored with the frame. A parser instruction start vector 422 is provided from the frame memory controller 424 to the instruction memory controller 460. The instruction memory controller 460 controls the instruction memory based upon the frame data from the frame memory controller 424.

The instruction memory 430 contains both instructions for the parsing engine 440 and layer offsets to access selected words in the dual port memory buffer 416. Each protocol layer within each frame is identified and its location in the memory buffer is stored in a table. The layer offset is added to the layer base address resulting in a memory pointer 426 which selects frame data 434 used for parsing. As each instruction from the instruction memory 430 is processed, the pointers are resolved by linking the layer address table 650 and layer offsets. The instruction pipe 432 and data pipe 434 serve to align the frame data with the instruction words.

The parsing engine 440 receives the instruction 436 and data 438 and performs the functions, e.g., relative compare 482, fixed compare 484, mask 486, register 488, and ALU 490, selected by the opcode in the parser instruction 436. The parser instruction set 436 defines the microcode used by the parsing engine 440. Branching 474 is provided to allow the instruction memory controller 460 to instruct the instruction memory 430 to generate instructions for the parsing engine 440 and layer offsets to access selected words in the dual port memory buffer 416. The branching 474 may be conditional branching, wherein a field in a packet is analyzed and instructions are executed based upon whether the field meets a predetermined condition.

The results 442 of the functions are passed to the extractor 444 which builds queries (search words) 446 to be search against either the CAM 448 or indexed directly to the SRAM 450. These results contain the frame classification or can be fed back to the instruction memory controller 460 to be used as either the start vector for subsequent processing stages or as data the parsing engine 440 can parse through the data pipe 434. When processing is complete, the frame is read out 470 and forwarded with the CAM/SRAM data, i.e., the search results 472. The search results 472 are provided as an output, for example, as a start instruction pointer for the editor, as a connection ID for policing, or as a different type of instruction for the editor such as for filtering.

The parsing engine 440 may also provide an update 480 to the search results 472, for example, to provide a pointer and other bits that the editor can interpret, to perform custom bit modifications, or to provide for a more dynamic configuration. The parsing engine 440 executes a new instruction each clock cycle. Because of the preprocessing by the preprocessor 410 , the blueprint of the frame is known before execution of parsing instructions is initiated.

In addition, the detailed block diagram 400 may be concatenated to process the incoming frame/packet input 412 to provide scalability of multi-protocol, multi-stage, real-time frame classification. The outputs 470, 472 are interleaved and are not all valid at the same time. By concatenating the block diagram 400, the bandwidth provided may be scaled to meet the needs of the network.

Fig. 5 illustrates a flow chart 500 representing the processes performed by the apparatus for providing multi-protocol, multi-stage, real-time frame classification. First a frame is received as input 510. The frame is provided to the preprocessor for classification 512. The frame is then stored in the memory buffer 514 as described above. The protocol layer code  is used as a start vector into the instruction memory 516. Extraction instructions are provided for building search queries 520. A decision is made whether there are additional searches to perform 522. If yes 524, then additional extraction instructions are provided for building search queries. If not 526, then the processing continues. A search latency is provided 530. A decision is made whether multi-stage parsing is to be performed 540. If yes 542, then the search results are fed back to the instruction controller to be used as the start vector for subsequent processing stages. If not 544, the frame and the search results are provided as an output 550.

Fig. 7 illustrates a flow chart 700 showing the processes of the detailed block diagram of the apparatus for providing multi-protocol, multi-stage, real-time frame classification with reference to Fig. 4 according to the present invention. First, preliminary multi-protocol frame classification for incoming frames is performed by resolving a composition for the incoming frame 710. Then, at least the incoming frame and the preliminary multi-protocol frame classification is processed to provide parsing instructions by storing an incoming frame in a memory buffer, generating the parsing instructions for the incoming frame based upon the multi-protocol frame classification and aligning the generated parsing instructions with a frame selected from the incoming frames stored in the memory buffer 712. At least a portion of the incoming frame is read from the memory buffer 714. The parsing instructions are interpreted 716. Macros are executed for performing parsing functions on at least the portion of the incoming frame using the interpreted parsing instructions 718. The parsing instructions may use conditional branching. Conditional branching involves analyzing a field in a packet and executing instructions based upon whether the field meets a predetermined condition. For example, if a field in the packet meets a predetermined criteria, a first instruction is performed. Otherwise, a second instruction is performed.

Extractions instructions are generated in response to the at least incoming frame and the preliminary multi-protocol frame classification 722. The extraction instructions may invoke conditional extractions. The extraction instructions are used to produce search queries 724. The search queries are applied against a memory to produce the search results 726. The search results may be used by the instruction memory controller 460 and/or frame memory controller 424 to generated further refined parsing instructions 740. The refined parsing instructions provide greater information about the incoming frame.

The process illustrated with reference to Fig. 7 may be tangibly embodied in a computer-readable medium or carrier, e.g. one or more of the fixed and/or removable data storage devices 260 illustrated in Fig. 2, or other data storage or data communications devices. The computer program 262 may be loaded into the classifier 210 or into the memory 244 to configure the classifier 210 of Fig. 2, for execution. The computer program 262 comprise instructions which, when read and executed by the classifier 210 of Fig. 2, causes the classifier 210 to perform the steps necessary to execute the steps or elements of the present invention.

The foregoing description of the exemplary embodiment of the invention has been presented for the purposes of illustration and description. It is not intended to be exhaustive or to limit the invention to the precise form disclosed. Many modifications and variations are possible in light of the above teaching. It is intended that the scope of the invention be limited not with this detailed description, but rather by the claims appended hereto.